



# SystemC CCI WG

## User Defined Data Type

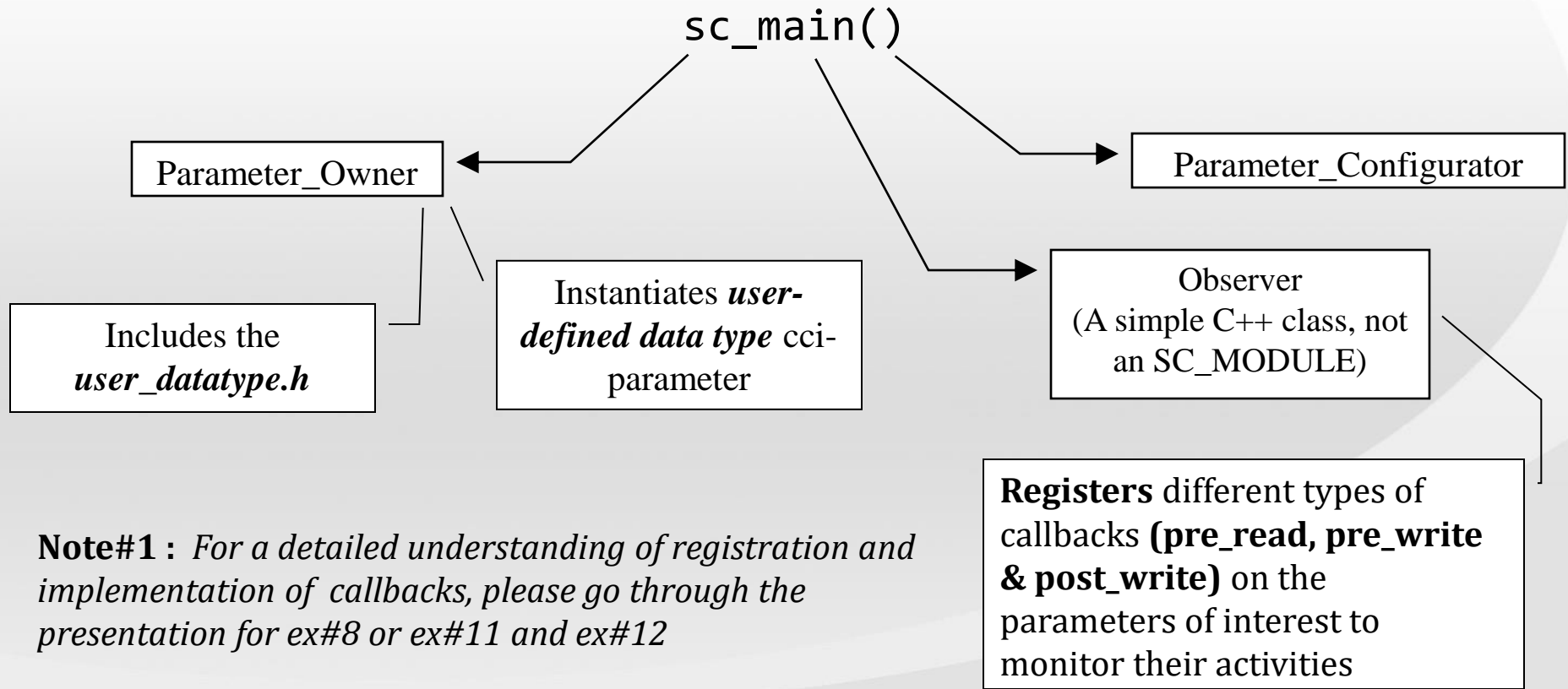
Girish Verma, P V S Phaneendra, CircuitSutra  
Technologies Pvt. Ltd, September 2011



# User Defined Data Types

- **Objective of the present example is to demonstrate the following :**
  - Adding support for user-defined data types, including complex structures
- **Show access using both JSON values and cci\_value**

# User Defined Data Types



**Note#1 :** *For a detailed understanding of registration and implementation of callbacks, please go through the presentation for ex#8 or ex#11 and ex#12*

### Within the Callbacks implementation :

Name of the originator : *ev.originator.name()*;

**Note#2 :** *The originator information will be displayed by the shape on the right*

```
Reporter : Observer
Callback Type : cb_type
Originator : originator_name
```

# User Defined Data Types

## Step#1 : *Defining a User-Defined Data Type*

Declare a C++ class/structure for the *user-defined data type* (see the following slide)

Create a default constructor for the user-defined data type structure

Also provide an overloaded constructor that is useful to assign the user-defined values in the desired manner

Implement `cci_value_converter<user-defined-type>` with pack and unpack functions (not shown – see example code)

Overload the insertion (`<<`) operator to provide convenience in reporting

# User Defined Data Types

## Infrastructure to be created : Creating Data Structure

```
struct route_table_ut
{
    // Default Constructor
    route_table_ut()
    : s_address(0x0)    // Source Address
      , d_address(0x0)  // Destination Address
      , index(0x0)      // Index
    {
        // Nothing to implement
    }

    // Overloaded Constructor
    route_table_ut(int saddr, int daddr, int idx)
    : s_address(saddr)
      , d_address(daddr)
      , index(idx)
    {
        // Nothing to implement
    }

    int s_address; // Slave Address
    int d_address; // Destination Address
    int index;     // Index
}; // struct route_table_ut
```

# User Defined Data Types

## Overloading *insertion* operator of C++

```
std::ostream& operator <<(std::ostream& os, const route_table_ut& ud)
{
    cci::cci_value udv(ud);
    return os << udv;
}
```

# User Defined Data Types

sc\_main()

Instantiate a cci\_originator instance to get access to the DEFAULT BROKER  
`cci::cci_originator myOriginator("sc_main_originator");`

Get handle to the DEFAULT BROKER using the above Originator

`cci::cci_broker_handle globalBroker =  
cci::cci_broker_manager::get_broker(myOriginator);`

Set preset value using `set_preset_value` API

`std::string init_str("{\"s_address\":\"256\",\"d_address\":\"512\",\"index\":\"0\"});  
globalBroker.set_preset_cci_value("param_owner.User_data_type_param",  
cci::cci_value::from_json(init_str));`

*Before the instantiation of the modules*

Parameter Owner

Parameter Configurator

Create instances of the modules now

`parameter_owner param_owner("param_owner");  
parameter_configurer param_cfgr("param_cfgr");  
observer observer_obj;`

# User Defined Data Types

## Parameter Owner

```
cci::cci_param<route_table_ut> udt_param;
```

Set name and assign default value:

```
udt_param("User_data_type_param",route_table_ut(0x200,0x300,1))
```

*Prior to 0ns*

Query parameter name:

```
udt_param.name();
```

Returns Name : *param\_owner.User\_data\_type\_param*

Query parameter's **Default Value** using **get\_default\_value()** API:

```
udt_param.get_default_value();
```

Returns Value = {"s\_address":512,"d\_address":768,"index":1}

Query parameter's **Value** using **get()** API:

```
udt_param.get();
```

Returns Value = {"s\_address":256,"d\_address":512,"index":0}

Set documentation :

```
const std::string init_desc = "This is user-defined data type";
```

```
udt_param.set_description(init_desc);
```

## Parameter Configurator

Declare handle for the parameter

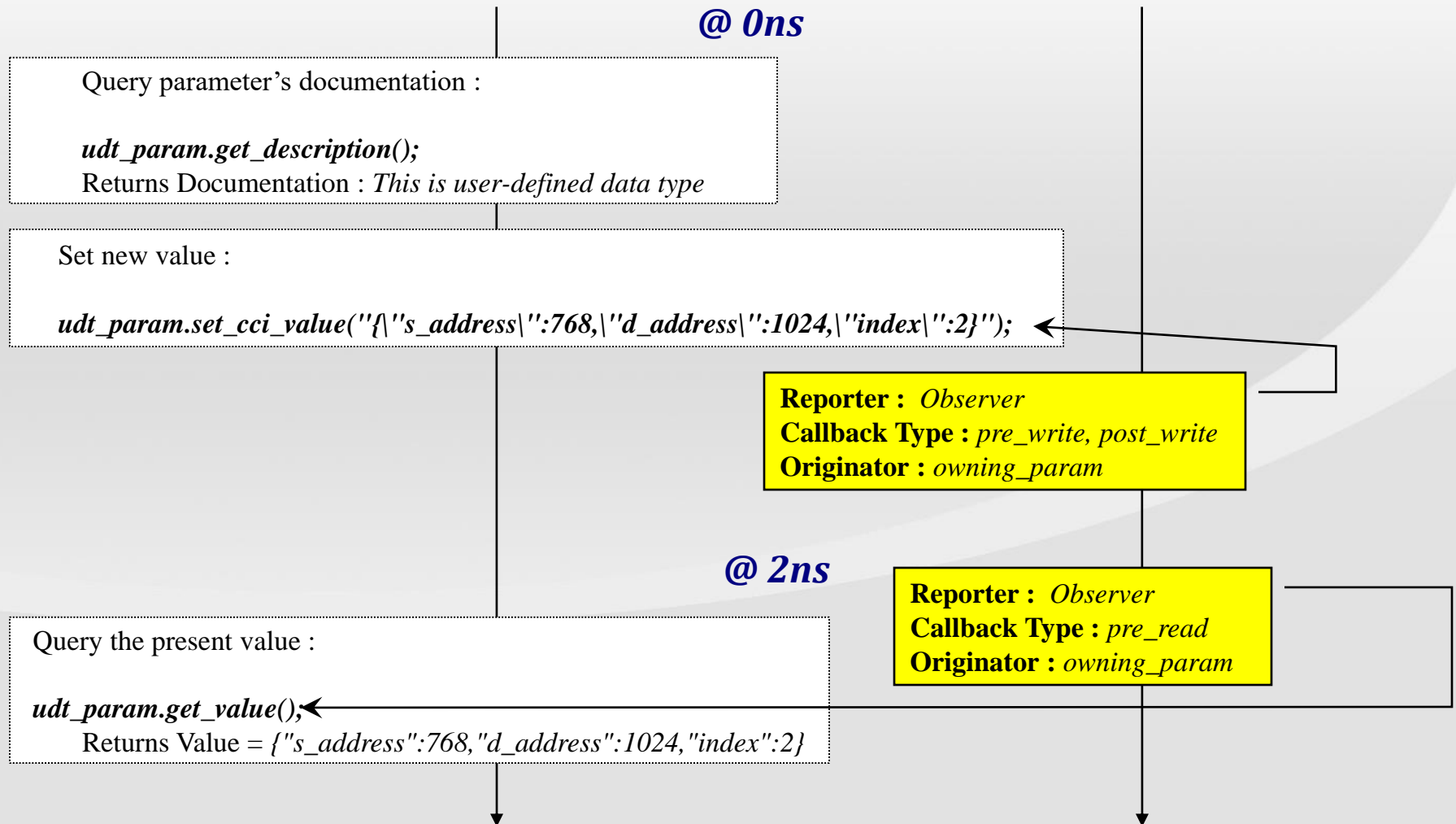
```
cci::cci_param_handle udt_param_handle;
```



# User Defined Data Types

## Parameter Owner

## Parameter Configurator



# User Defined Data Types

## Parameter Owner

## Parameter Configurator

@ 4ns

Query Name : *udt\_param\_handle.name()*;  
Returns value = *param\_owner.User\_data\_type\_param*

Query Value : *udt\_param\_handle.get\_cci\_value()*;  
Returns Value : {"s\_address":768,"d\_address":1024,"index":2}

Query Documentation : *udt\_param\_handle.get\_description()*;  
Returns Value : *This is user-defined data type*

@ 6ns

Set New Value :  
*std::string set\_string "{ \"s\_address\":1024, \"d\_address\":1280, \"index\":3 }";*  
*udt\_param\_handle.set\_cci\_value(cci::cci\_value::from\_json(set\_string));*

**Reporter : Observer**  
**Callback Type : pre\_read**  
**Originator : param\_cfgr**

**Reporter : Observer**  
**Callback Type : pre\_write, post\_write**  
**Originator : param\_cfgr**

# User Defined Data Types

Parameter Owner

Parameter Configurator

*@ 8ns*

Query Name : *udt\_param\_handle.name();*  
Returns value = *param\_owner.User\_data\_type\_param*

Query Value : *udt\_param\_handle.get\_cci\_value();*  
Returns Value : *{"s\_address":1024,"d\_address":1280,"index":3}*

**Reporter : *Observer***  
**Callback Type : *pre\_read***  
**Originator : *param\_cfgr***

# Expected Output

## (ex16\_User\_Defined\_Data\_Type.log)

SystemC Simulation

Info: sc\_main: [MAIN] : Setting preset value 's\_address:256,d\_address:512,index:0' to UDT

Info: param\_owner: @0 s, Prior to 0 s demonstrating 'get\_default\_value()'

Info: param\_owner: @0 s, [OWNER -> Retrieve] : Parameter name : param\_owner.User\_data\_type\_param

Info: param\_owner: @0 s, [OWNER -> Retrieve] : Using 'get\_default\_value()' :  
{ "s\_address":512,"d\_address":768,"index":1 }

Info: param\_owner: @0 s, [OWNER -> Retrieve] : Parameter Value' :  
{ "s\_address":256,"d\_address":512,"index":0 }

Info: param\_owner: @0 s, [OWNER -> Set] : Param desc - 'This is user-defined data type

Info: param\_cfgr: @0 s, [CFGR C\_TOR] : Broker Type : DEFAULT\_BROKER - is not a private broker.

[OBSERVER C\_TOR] : Broker Type : DEFAULT\_BROKER - is not a private broker.

Info: sc\_main: Begin Simulation.

Info: param\_owner: @0 s, @ 0 s

Info: param\_owner: @0 s, [OWNER -> Retrieve] : Description : This is user-defined data type

Info: param\_owner: @0 s, [OWNER -> Set] : New Value param\_owner.User\_data\_type\_param  
's\_address:768,d\_address:1024,index:2'

[OBSERVER pre\_write\_cb] : Parameter Name : param\_owner.User\_data\_type\_paramOriginator info  
: param\_owner

# Cont'd

[OBSERVER post\_write\_cb] : Parameter Name : param\_owner.User\_data\_type\_param  
Originator info : param\_owner

Info: param\_owner: @2 ns, @ 2 ns

[OBSERVER pre\_read\_cb] : Parameter Name : param\_owner.User\_data\_type\_param Originator info : param\_owner

[OBSERVER post\_read\_cb]: Parameter Name : param\_owner.User\_data\_type\_param Originator info : param\_owner

Info: param\_owner: @2 ns, [OWNER -> Retrieve] : UDT Value :  
{ "s\_address":768,"d\_address":1024,"index":2}

Info: param\_cfgr: @4 ns, @ 4 ns

Info: param\_cfgr: @4 ns, [CFGR -> Retrieve] : Parameter name : param\_owner.User\_data\_type\_param

[OBSERVER pre\_read\_cb] : Parameter Name : param\_owner.User\_data\_type\_param Originator info : param\_cfgr

[OBSERVER post\_read\_cb]: Parameter Name : param\_owner.User\_data\_type\_param Originator info : param\_cfgr

Info: param\_cfgr: @4 ns, [CFGR -> Retrieve] : Parameter value:  
{ "s\_address":768,"d\_address":1024,"index":2}

Info: param\_cfgr: @4 ns, [CFGR -> Retrieve] : Parameter desc: This is user-defined data type

# Cont'd

Info: param\_cfgr: @6 ns, @ 6 ns

Info: param\_cfgr: @6 ns, [CFGR -> Set] : Value - 's\_address:1024,d\_address:1280,index:3

[OBSERVER pre\_write\_cb] : Parameter Name : param\_owner.User\_data\_type\_param Originator info : param\_cfgr

[OBSERVER post\_write\_cb] : Parameter Name : param\_owner.User\_data\_type\_param Originator info : param\_cfgr

Info: param\_cfgr: @8 ns, @ 8 ns

Info: param\_cfgr: @8 ns, [CFGR -> Retrieve] : Parameter name : param\_owner.User\_data\_type\_param

[OBSERVER pre\_read\_cb] : Parameter Name : param\_owner.User\_data\_type\_param Originator info : param\_cfgr

[OBSERVER post\_read\_cb]: Parameter Name : param\_owner.User\_data\_type\_param Originator info : param\_cfgr

Info: param\_cfgr: @8 ns, [CFGR -> Retrieve] : Parameter value:  
{"s\_address":1024,"d\_address":1280,"index":3}

Info: sc\_main: End Simulation.